F 5

```
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBB         BBB  OOO         OOO  OOO         OOO        TTT        SSS
BBB         BBB  OOO         OOO  OOO         OOO        TTT        SSS
BBB         BBB  OOO         OOO  OOO         OOO        TTT        SSS
BBB         BBB  OOO         OOO  OOO         OOO        TTT        SSS
BBB         BBB  OOO         OOO  OOO         OOO        TTT        SSS
BBB         BBB  OOO         OOO  OOO         OOO        TTT        SSS
BBBBBBBBBBBB     OOO         OOO  OOO         OOO        TTT           SSSSSSSSS
BBBBBBBBBBBB     OOO         OOO  OOO         OOO        TTT           SSSSSSSSS
BBBBBBBBBBBB     OOO         OOO  OOO         OOO        TTT           SSSSSSSSS
BBB         BBB  OOO         OOO  OOO         OOO        TTT                    SSS
BBB         BBB  OOO         OOO  OOO         OOO        TTT                    SSS
BBB         BBB  OOO         OOO  OOO         OOO        TTT                    SSS
BBB         BBB  OOO         OOO  OOO         OOO        TTT                    SSS
BBB         BBB  OOO         OOO  OOO         OOO        TTT                    SSS
BBB         BBB  OOO         OOO  OOO         OOO        TTT                    SSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO           TTT        SSSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO           TTT        SSSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO           TTT        SSSSSSSSSSSS
```

```
                                                  M  3
**FILE**ID**WRITEBOOT




WW      WW  RRRRRRRR      IIIIII    TTTTTTTTTT  EEEEEEEEEE  BBBBBBBB      OOOOOO      OOOOOO    TTTTTTTTTT
WW      WW  RRRRRRRR      IIIIII    TTTTTTTTTT  EEEEEEEEEE  BBBBBBBB      OOOOOO      OOOOOO    TTTTTTTTTT
WW      WW  RR      RR      II          TT      EE          BB      BB  OO      OO  OO      OO      TT
WW      WW  RR      RR      II          TT      EE          BB      BB  OO      OO  OO      OO      TT
WW      WW  RR      RR      II          TT      EE          BB      BB  OO      OO  OO      OO      TT
WW      WW  RR      RR      II          TT      EE          BB      BB  OO      OO  OO      OO      TT
WW      WW  RRRRRRRR        II          TT      EEEEEEEE    BBBBBBBB    OO      OO  OO      OO      TT
WW      WW  RRRRRRRR        II          TT      EEEEEEEE    BBBBBBBB    OO      OO  OO      OO      TT
WW  WW  WW  RR  RR          II          TT      EE          BB      BB  OO      OO  OO      OO      TT
WW  WW  WW  RR  RR          II          TT      EE          BB      BB  OO      OO  OO      OO      TT
WWWW  WWWW  RR    RR        II          TT      EE          BB      BB  OO      OO  OO      OO      TT        ....
WWWW  WWWW  RR    RR        II          TT      EE          BB      BB  OO      OO  OO      OO      TT        ....
WW      WW  RR      RR    IIIIII        TT      EEEEEEEEEE  BBBBBBBB      OOOOOO      OOOOOO        TT        ....
WW      WW  RR      RR    IIIIII        TT      EEEEEEEEEE  BBBBBBBB      OOOOOO      OOOOOO        TT        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE writeboot (      ! Writes boot block code and data into LBN 0
0                         IDENT = 'V04-000',
0                         MAIN = write_boot
0                         ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1 !++
1 ! FACILITY:
1 !
1 !       WRITEBOOT
1 !
1 ! ABSTRACT:
1 !
1 !       The purpose of this utility is to write a BOOTable program into
1 !       LBN 0 of a system disk or TU58.  This BOOTable program will
1 !       contain within its first three longwords, the starting LBN and
1 !       size of a primary VMS bootstrap file located on this same system
1 !       disk or TU58 and also the relative location in memory where the
1 !       primary bootstrap should be loaded.  The system disk or TU58 may be
1 !       a FILES11 (ODS-2) or an RT-11 formatted device.
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX/VMS operating system, requires LOG_IO privilege. Assumes
1 !       bootstrap file is VMB.EXE unless otherwise specified.
1 !
1 ! AUTHOR:
1 !
1 !       Carol Peters    20 June 1979
1 !
1 ! REVISION HISTORY:
1 !
1 !       V03-003 TCM0001         Trudy C. Matthews       10-Aug-1983
```

```
1 !                     Ensure that the bootfile is contiguous before writing the
1 !                     boot block.
1 !
1 !             V03-002 RAS0175         Ron Schaefer            28-Jul-1983
1 !                     Eliminate useless reference to FAB$V_UFM.
1 !
1 !             V03-001 PCA1006         Paul C. Anagnostopoulos  9-Dec-1982
1 !                     Modify the initialization of the BOOTBLOCK.EXE RMS control
1 !                     blocks so that the logical name BOOTBLOCK can be used.
1 !                     On systems with library disks, this image is not on
1 !                     the system disk.
1 !
1 !             V02-002 STJ0054         Steven T. Jeffreys,             29-Jun-1981
1 !                     Changed external routine references to use general addressing mode.
1 !
1 !             Robert Rappaport 10 Aug 1979
1 !                     Major changes to accomodate RT-11 format devices.
1 !
1 !             Steve Jeffreys   12 Nov 1979
1 !                       - Enable WRITEBOOT to accept input from an indirect command file.
1 !                       - Remove verify prompt for target device.
1 !                       - Add prompt for VBN of code in boot file.
1 !                       - Allow boot code to be loaded at an arbitrary address.
1 !--
```

```
1 !
1 ! Table of contents
1 !
1
1 FORWARD ROUTINE
1         write_boot;
1
1 !
1 ! Include files
1 !
1
1 LIBRARY 'SYS$LIBRARY:LIB.L32';                       ! VMS system definitions.
1
1 !
1 ! External declarations
1 !
1
1 EXTERNAL ROUTINE
1         ots$cvt_tz_l     : ADDRESSING_MODE (GENERAL),
1         lib$index        : ADDRESSING_MODE (GENERAL),
1         lib$put_output   : ADDRESSING_MODE (GENERAL),
1         lib$sfree1_dd    : ADDRESSING_MODE (GENERAL),
1         RTF$TARGET_DEV   : ADDRESSING_MODE (GENERAL),
1         RTF$OPENFILE     : ADDRESSING_MODE (GENERAL),
1         lib$get_input    : ADDRESSING_MODE (GENERAL);
1
1 !
1 ! Macros
1 !
1
1 MACRO
1         bbl_l_filesize  = 0,0,32,0%,           ! Primary boot file size in blocks.
1         bbl_w_hiordlbn  = 4,0,16,0%,           ! Hi order word of starting LBN of primary boot
1         bbl_w_loordlbn  = 6,0,16,0%,           ! Low order word of starting LBN of primary boot
1         bbl_l_loadadr   = 8,0,32,0%;           ! Address at which to load primary boot file
1                                                ! (expressed as offset from sp).
1
1 !
1 ! Own storage
1 !
1
1 OWN
1         priboo_descrip  : BLOCK [8, BYTE] INITIAL      ! Device/file spec descriptor.
1                           (BYTE
1                                   (REP 3 OF (0),
1                                   dsc$k_class_d,
1                                   REP 4 OF (0))),
1         loadadr_descrip : BLOCK [8, BYTE] INITIAL      ! Load address string descriptor.
1                           (BYTE
1                                   (REP 3 OF (0),
1                                   dsc$k_class_d,
1                                   REP 4 OF (0))),
1         prompt_descrip  : BLOCK [8, BYTE] INITIAL      ! Prompt string descriptor.
1                           (BYTE
1                                   (REP 3 OF (0),
1                                   dsc$k_class_s)),
1
```

```
;  140      0139  1        vbn_descrip     : BLOCK [8, BYTE] INITIAL        ! Prompt string for VBN
;  141      0140  1                           (BYTE
;  142      0141  1                                  (REP 3 OF (0),
;  143      0142  1                                  dsc$k_class_d,
;  144      0143  1                                  REP 4 OF (0))),
;  145      0144  1
;  146      0145  1        priboo_fab      : $FAB_DECL,                     ! Primary bootstrap file's FAB.
;  147      0146  1        bootbl_fab      : $FAB_DECL,                     ! Boot block file's FAB.
;  148      0147  1
;  149      0148  1        priboo_filnam   : VECTOR [nam$c_maxrss, BYTE],   ! Primary bootstrap file name after open.
;  150      0149  1        priboo_exp_name : VECTOR [nam$c_maxrss, BYTE],   ! Primary bootstrap file name before open.
;  151      0150  1
;  152      0151  1 !      bootbl_filnam   : VECTOR [nam$c_maxrss, BYTE],   ! Boot block file name after open.
;  153      0152  1 !      bootbl_exp_name : VECTOR [nam$c_maxrss, BYTE],   ! Boot block file name before open.
;  154      0153  1
;  155      0154  1 !      result_nam_blk  : $NAM (RSA = priboo_filnam),    ! Related file NAM block.
;  156      0155  1
;  157      0156  1 !      bootbl_nam_blk  : $NAM (                         ! Name block for BOOTBLOCK.EXE.
;  158      0157  1 !                           RSA = bootbl_filnam,
;  159      0158  1 !                           RSS = nam$c_maxrss,
;  160      0159  1 !                           ESA = bootbl_exp_name,
;  161      0160  1 !                           ESS = nam$c_maxrss,
;  162      0161  1 !                           RLF = result_nam_blk),
;  163      0162  1
;  164    P 0163  1        priboo_nam_blk  : $NAM (                         ! Name block for primary bootstrap.
;  165    P 0164  1                           RSA = priboo_filnam,
;  166    P 0165  1                           RSS = nam$c_maxrss,
;  167    P 0166  1                           ESA = priboo_exp_name,
;  168      0167  1                           ESS = nam$c_maxrss),
;  169      0168  1
;  170      0169  1        priboo_xabfhc   : $XABFHC (),                    ! Primary bootstrap file header characteristics bloc
;  171      0170  1 !      bootbl_xabfhc   : $XABFHC (),                    ! Boot block file header characteristics block.
;  172      0171  1
;  173      0172  1        privilege_mask  : BLOCK [8, BYTE],
;  174      0173  1        getjpi_itemlist : BLOCK [4, LONG] INITIAL
;  175      0174  1                           (WORD (8, JPI$_PROCPRIV),
;  176      0175  1                           LONG (privilege_mask, 0,0)),
;  177      0176  1        io_stat_block   : VECTOR [2],
;  178      0177  1
;  179      0178  1        two_block_buf   : BLOCK [1024, BYTE],
;  180      0179  1
;  181      0180  1        bootdev_descrip : BLOCK [8, BYTE],
;  182      0181  1        bootdev_chan    : WORD,
;  183      0182  1        load_adr                : LONG,
;  184      0183  1        devchar_buff    : BLOCK [3, LONG]       ! Buffer to receive device characteristics.
;  185      0184  1                          INITIAL (LONG (REP 3 OF (0))),
;  186      0185  1
;  187      0186  1
;  188      0187  1        devchar_descrip : BLOCK [8, BYTE]       ! Descriptor for Device characteristics.
;  189      0188  1                          INITIAL (LONG (12, devchar_buff)),
;  190      0189  1
;  191      0190  1        filnam_descrip  : BLOCK [8, BYTE]                ! Descriptor for just the file name (no device or di
;  192      0191  1                          INITIAL (BYTE (REP 8 OF (0))),
;  193      0192  1        filspec_descrip : BLOCK [8, BYTE]                ! Descriptor for file spec returned from $PARSE.
;  194      0193  1                          INITIAL (LONG (0, priboo_exp_name)),
;  195      0194  1
;  196      0195  1        vbn             : VECTOR [1, LONG],              ! VBN of boot file code
```

```
1          stat_block      : VECTOR [2, LONG];             ! Area to hold LBN and size of specified file.
1
1  BIND
1          block_buffer = two_block_buf : BLOCK [512, BYTE],
1          logio_msg = UPLIT BYTE (%ASCII 'You lack LOG_IO privilege.') : VECTOR [, LONG],
1          vbn_bnds_msg = UPLIT BYTE (%ASCII 'VBN must be >= 1.') : VECTOR [, LONG],
1          notcontig_msg = UPLIT BYTE (%ASCII 'Boot file is not contiguous.')
1                                          : VECTOR [, LONG],
1          remount_msg = UPLIT BYTE (%ASCII 'You lack READ and/or WRITE access to TARGET DEVICE.  DISMOUNT and
1                                  : VECTOR [, LONG],
1          ascii_bracket = UPLIT BYTE (%ASCII ']') : VECTOR [, LONG],
1          prompt_buffer = UPLIT BYTE (%ASCII 'Target system device (and boot file if not VMB.EXE): ') : VECTOR
1          prompt2_buffer = UPLIT BYTE (%ASCII 'Enter load address of primary bootstrap in HEX (default is 200)
1          prompt3_buffer = UPLIT BYTE (%ASCII 'Enter VBN of boot file code (default is 1) : ') : VECTOR [, LON
1          priboo_def_name = UPLIT BYTE (%ASCII '[SYSEXE]VMB.EXE') : VECTOR [, LONG];
1
1  LITERAL
1          dev_offset       = 18,
1          logio_length     = 26,
1          vbn_bnds_len     = 17,
1          remount_length   = 77,
1          notcontig_length= 28,
1          prompt_length    = 53,                     ! Length of prompt.
1          prompt2_length   = 65,                     ! Length of prompt2.
1          prompt3_length   = 45,                     ! Length of prompt3.
1          bootname_length  = 15;                     ! Length of VMB.EXE.
1
1  OWN
1          yes_no_buf       : BLOCK  [1, BYTE] INITIAL (BYTE (0)),
1          yes_no_descrip   : BLOCK  [8, BYTE]
1                                   INITIAL (LONG (1, yes_no_buf)),
1
1          logio_descrip    : BLOCK [8, BYTE]
1                                   INITIAL (LONG (logio_length, logio_msg)),
1          vbn_bnds_descrip: BLOCK [8,BYTE]
1                                   INITIAL (LONG (vbn_bnds_len, vbn_bnds_msg)),
1          remount_descrip : BLOCK [8, BYTE]
1                                   INITIAL (LONG (remount_length, remount_msg)),
1          notcontig_descrip
1                           : BLOCK [8,BYTE]
1                                   INITIAL (LONG (notcontig_length, notcontig_msg)),
1          bracket_descrip : BLOCK [8, BYTE]               ! Descriptor for constant string consisting of "]".
1                                 INITIAL (LONG (1, ascii_bracket));
1
```

```
1 ROUTINE write_boot =    ! Writes the boot block.
1
1 !
1 ! Functional description:
1 !
1 !       1. Prompts for target system device and optional boot file spec.
1 !
1 !       2. Determines if target device is Files-11 or FOREIGN.
1 !
1 !       3. Determines starting LBN and size of VMB.EXE (or specified file)
1 !          on the target system device specified by the user in step #1.
1 !               In case of Files-11 device this means opening file with
1 !                       an XABFHC specified in the FAB.
1 !               In case of a FOREIGN device this means calling external
1 !                       routine "RTF$OPENFILE".
1 !
1 !       -  Prompt for VBN of boot file code.
1 !
1 !       4. Prompts for memory location where primary bootstrap should be
1 !          loaded in memory.
1 !
1 !       5. Opens SYS$SYSTEM:BOOTBLOCK.EXE on the current system disk.
1 !
1 !       6. Reads VBN #0 of SYS$SYSTEM:BOOTBLOCK.EXE into buffer.
1 !
1 !       7. Modifies buffer by placing starting LBN, size and memory location
1 !          obtained in steps #3 and #4 above, into the buffer at the
1 !          appropriate places.
1 !
1 !       8. Writes buffer containing modified copy of SYS$SYSTEM:BOOTBLOCK.EXE
1 !          into LBN #0 of target system device specified by user in step #1.
1 !
1 !       9. Closes files.
1 !
1 ! Inputs:
1 !
1 !       none
1 !
1 ! Outputs:
1 !
1 !       R0 contains a status code.
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       index,
2       status;
2
2 ! Issue a $GETJPI system service call to discover whether the process
2 ! executing WRITEBOOT has LOG_IO privilege. If not, don't allow process
2 ! to write on the target system disk.
2 !
2
3 IF NOT (status = $getjpi (efn = 3, itmlst = getjpi_itemlist, iosb = io_stat_block))
2 THEN RETURN .status;
```

```
     2  IF NOT .privilege_mask [prv$v_log_io]
     3  THEN BEGIN
     3          lib$put_output (logio_descrip);
     3          RETURN SS$_NOPRIV;
     2       END;
     2
     2  !
     2  ! Prompt for the target system device name optionally followed by the
     2  ! name of a primary bootstrap file.
     2  !
     2
     2  prompt_descrip [dsc$w_length] = prompt_length;
     2  prompt_descrip [dsc$a_pointer] = prompt_buffer;
     2
     2  IF .priboo_descrip [dsc$w_length] NEQ 0
     2          THEN lib$sfree1_dd (priboo_descrip);    ! deallocate previous string.
     2
     2  WHILE .priboo_descrip [dsc$w_length] EQL 0
     2  DO
     3      BEGIN
     3      status = lib$get_input (priboo_descrip, prompt_descrip);
     3      IF NOT .status
     3      THEN RETURN .status;
     2      END;
     2
     2  !
     2  ! Translate all lower case alphabetic characters to upper case so that
     2  ! an RMS translation will work.
     2  !
     2
     3  INCR count FROM 0 TO (.priboo_descrip [dsc$w_length] - 1)
     2  DO
     3      BEGIN
     3      BIND
     3          file_spec = .priboo_descrip [dsc$a_pointer] : VECTOR [, BYTE];
     4      IF ((.file_spec [.count] GEQ 'a') AND (.file_spec [.count] LEQ 'z'))
     3      THEN file_spec [.count] = .file_spec [.count] - %x'20';
     2      END;
     2
     2  !
     2  ! Determine if the target device is Files-11 or FOREIGN.  Do this
     2  ! by $PARSEing the given file spec using the default of [SYSEXE]VMB.EXE
     2  ! and by specifying a NAM block.  With NAM block we obtain the device
     2  ! name in the nam$t_dvi field and we build a string descriptor for this
     2  ! string and use system service $GETDEV to get the device characteristics.
     2  !
     2
     2
P    2  $FAB_INIT (
P    2                          FAB = priboo_fab,
P    2                          FAC = <GET>,
P    2                          FNA = .priboo_descrip [dsc$a_pointer],
P    2                          FNS = .priboo_descrip [dsc$w_length],
P    2                          DNA = priboo_def_name,
P    2                          DNS = bootname_length,
P    2                          FOP = <NAM>,
P    2                          NAM = priboo_nam_blk,
  2                 XAB = priboo_xabfhc);
  3 IF NOT (status = $PARSE (FAB = priboo_fab))
  2     THEN RETURN .status;
  2
  2 bootdev_descrip[dsc$w_length] = .(priboo_nam_blk[nam$t_dvi]) <0,8>;
  2 bootdev_descrip[dsc$a_pointer] = priboo_nam_blk[nam$t_dvi] + 1;
  2
  2
P 3 IF NOT (status = $GETDEV (DEVNAM = bootdev_descrip,
  3                           PRIBUF = devchar_descrip))
  2     THEN RETURN .status;
  2
  2
  2 !
  2 ! At this point we have the target device characteristics.  If the
  2 ! device is FOREIGN then we isolate the file name in the expanded
  2 ! file spec and build a string descriptor for this substring.
  2 ! Next we call RTF$TARGET_DEV to record the name of the target device.
  2 ! Then we call RTF$OPENFILE to get the starting LBN and size.  If
  2 ! on the other hand the device is Files-11, then we simply open the file.
  2 ! The purpose of the open is to load the size and starting LBN of the
  2 ! file into the XABFHC block produced by RMS.  In this latter case of a
  2 ! Files-11 device we then copy this data out of the XABFHC block into
  2 ! the OWN variable stat_block.
  2 !
  2
  2 IF .devchar_buff[dev$v_for]                                    ! i.e. if FOREIGN
  3     THEN BEGIN
  3             filspec_descrip[dsc$w_length] = .priboo_nam_blk[nam$b_esl];
  3             index = lib$index (filspec_descrip, bracket_descrip);
  3             filnam_descrip[dsc$a_pointer] = .filspec_descrip[dsc$a_pointer] + .index;
  3             filnam_descrip[dsc$w_length]  = .filspec_descrip[dsc$w_length]  - .index;
  3
  3             RTF$TARGET_DEV (bootdev_descrip);
  3
  4             IF NOT (status = RTF$OPENFILE (filnam_descrip,
  4                                            two_block_buf,
  4                                            stat_block))
  4                THEN BEGIN
  4                        lib$put_output (remount_descrip);
  4                        RETURN .status;
  3                     END;
  3          END
  3     ELSE BEGIN
  4            IF NOT (status = $RMS_OPEN (FAB = priboo_fab))
  3               THEN RETURN .status;
  3
  3             stat_block[0] = .priboo_xabfhc[xab$l_sbn];
  3             IF .priboo_xabfhc[xab$l_sbn] EQL 0
  4                THEN BEGIN
  4                        lib$put_output (notcontig_descrip);
  4                        $RMS_CLOSE (FAB = priboo_fab);
  4                        RETURN SS$_FILNOTCNTG;
  3                END;
  3              IF .priboo_xabfhc[xab$w_ffb] NEQ 0
  3                THEN stat_block[1] = .priboo_xabfhc[xab$l_ebk]
  3                ELSE stat_block[1] = .priboo_xabfhc[xab$l_ebk] - 1;
```

```
          $RMS_CLOSE (FAB = priboo_fab);
        END;


!
! Prompt the user for the VBN of the boot file code.
!

prompt_descrip[dsc$w_length]  = prompt3_length; ! Set up prompt descriptor
prompt_descrip[dsc$a_pointer] = prompt3_buffer;

status = 0;
WHILE NOT .status
DO
    BEGIN
    IF .vbn_descrip[dsc$w_length] NEQ 0
        THEN lib$sfree1_dd (vbn_descrip);        ! Deallocate previous string

    IF NOT (status = lib$get_input (vbn_descrip, prompt_descrip)) ! Prompt for VBN
        THEN RETURN .status;

    IF .vbn_descrip[dsc$w_length] NEQ 0          ! Convert string to decimal #
        THEN status = ots$cvt_tz_l (vbn_descrip,vbn)
        ELSE vbn = 1;                            ! Default VBN

    IF .vbn LSS 1                                ! Check for VBN < 1
        THEN
            BEGIN
            IF NOT (status = lib$put_output (vbn_bnds_descrip))
                THEN RETURN .status;
            status = 0;
            END;
    END;                                         ! End of VBN prompt WHILE loop

stat_block[0] = .stat_block[0] + (.vbn - 1);     ! Update LBN to point to boot code

!
! Open the bootblock file (called SYS$SYSTEM:BOOTBLOCK.EXE) located on the
! system disk.  Ensure that the logical name BOOTBLOCK will work.
!

$FAB_INIT (
                FAB = bootbl_fab,
                DNM = 'SYS$SYSTEM:.EXE',
                FAC = <BIO>,
                FNM = 'BOOTBLOCK',
                FOP = <UFO>);
IF NOT (status = $RMS_OPEN (FAB = bootbl_fab))
   THEN RETURN .status;

!
!
! Read the first block of BOOTBLOCK.EXE into a page-long buffer in
! memory.
!

IF NOT (status = $qiow (
                        CHAN = .bootbl_fab [fab$l_stv],
                        FUNC = io$_readvblk,
                        P1 = block_buffer,
                        P2 = 512,
                        P3 = 1))
THEN RETURN .status;

!
! Here we prompt the user for the relative memory location that he wants
! the primary bootstrap loaded into.
!

prompt_descrip[dsc$w_length] = prompt2_length;
prompt_descrip[dsc$a_pointer] = prompt2_buffer;

status = 0;     ! Set to false for following loop.

WHILE NOT .status
DO
BEGIN
        IF .loadadr_descrip[dsc$w_length] NEQ 0
                THEN lib$sfree1_dd (loadadr_descrip);

        status = lib$get_input (loadadr_descrip, prompt_descrip);
        IF NOT .status THEN RETURN .status;

        IF .loadadr_descrip[dsc$w_length] NEQ 0
         THEN status = ots$cvt_tz_l(loadadr_descrip, load_adr)
         ELSE load_adr = 512;                                   ! Default
END;

!
! Load the starting LBN, size and relative load location into the first
! 3 longwords of the buffer containing the BOOTBLOCK code.
!

block_buffer [bbl_l_filesize] = .stat_block[1];           ! Copy filesize.
block_buffer [bbl_w_hiordlbn] = .(stat_block[0])<16,16>;  ! Swap LBN words for
block_buffer [bbl_w_loordlbn] = .(stat_block[0])<0,16>;   ! DSC
block_buffer [bbl_l_loadadr] = .load_adr;                 ! Copy where to load
                                                  ! primary bootstrap


!
! Assign a channel to target device.
!

IF NOT (status = $assign (
                        DEVNAM = bootdev_descrip,
                        CHAN = bootdev_chan))
THEN RETURN .status;

!
! Copy the page-long buffer into LBN 0 of the target system device.
!

IF NOT (status = $qiow (
;  527       P 0525  3                      CHAN = .bootdev_chan,
;  528       P 0526  3                      FUNC = io$_writelblk,
;  529       P 0527  3                      P1 = block_buffer,
;  530       P 0528  3                      P2 = 512,
;  531         0529  3                      P3 = 0))
;  532         0530  2  THEN RETURN .status;
;  533         0531  2
;  534         0532  2  !
;  535         0533  2  ! Close the open files.
;  536         0534  2  !
;  537         0535  2
;  538         0536  2  $RMS_CLOSE (FAB = bootbl_fab);
;  539         0537  2
;  540         0538  2  !
;  541         0539  2  ! Return with success status.
;  542         0540  2  !
;  543         0541  2
;  544         0542  2  RETURN SS$_NORMAL;
;  545         0543  1  END;


                                                                                              .TITLE  WRITEBOOT
                                                                                              .IDENT  \V04-000\

                                                                                              .PSECT  $PLIT$,NOWRT,NOEXE,2

4F   49   5F   47   4F   4C   20   6B   63   61   6C   20   75   6F   59   00000 P.AAA:  .ASCII  \You lack LOG_IO privilege.\
                    2E   65   67   65   6C   69   76   69   72   70   20   0000F
20   3D   3E   20   65   62   20   74   73   75   6D   20   4E   42   56   0001A P.AAB:  .ASCII  \VBN must be >= 1.\
                                                                 2E   31   00029
6F   6E   20   73   69   20   65   6C   69   66   20   74   6F   6F   42   0002B P.AAC:  .ASCII  \Boot file is not contiguous.\
          2E   73   75   6F   75   67   69   74   6E   6F   63   20   74   0003A
61   20   44   41   45   52   20   6B   63   61   6C   20   75   6F   59   00047 P.AAD:  .ASCII  \You lack READ and/or WRITE access to TAR\
63   63   61   20   45   54   49   52   57   20   72   6F   2F   64   6E   00056
                         52   41   54   20   6F   74   20   73   73   65   00065
49   44   20   20   2E   45   43   49   56   45   44   20   54   45   47   0006F         .ASCII  \GET DEVICE.  DISMOUNT and reMOUNT it.\
4F   4D   65   72   20   64   6E   61   20   54   4E   55   4F   4D   53   0007E
                                        2E   74   69   20   54   4E   55   0008D
                                                                      5D   00094 P.AAE:  .ASCII  \]\
64   20   6D   65   74   73   79   73   20   74   65   67   72   61   54   00095 P.AAF:  .ASCII  \Target system device (and boot file if n\
74   6F   6F   62   20   64   6E   61   28   20   65   63   69   76   65   000A4
                         6E   20   66   69   20   65   6C   69   66   20   000B3
          20   3A   29   45   58   45   2E   42   4D   56   20   74   6F   000BD         .ASCII  \ot VMB.EXE): \
72   64   64   61   20   64   61   6F   6C   20   72   65   74   6E   45   000CA P.AAG:  .ASCII  \Enter load address of primary bootstrap \
20   79   72   61   6D   69   72   70   20   66   6F   20   73   73   65   000D9
                         20   70   61   72   74   73   74   6F   6F   62   000E8
74   6C   75   61   66   65   64   28   20   58   45   48   20   6E   69   000F2         .ASCII  \in HEX (default is 200): \
                         20   3A   29   30   30   32   20   73   69   20   00101
6F   62   20   66   6F   20   4E   42   56   20   72   65   74   6E   45   0010B P.AAH:  .ASCII  \Enter VBN of boot file code (default is \
64   28   20   65   64   6F   63   20   65   6C   69   66   20   74   6F   0011A
                         20   73   69   20   74   6C   75   61   66   65   00129
                                                  20   3A   20   29   31   00133         .ASCII  \1) : \
45   58   45   2E   42   4D   56   5D   45   58   45   53   59   53   5B   00138 P.AAI:  .ASCII  \[SYSEXE]VMB.EXE\
                              4B   43   4F   4C   42   54   4F   4F   42   00147 P.AAJ:  .ASCII  \BOOTBLOCK\
45   58   45   2E   3A   4D   45   54   53   59   53   24   53   59   53   00150 P.AAK:  .ASCII  \SYS$SYSTEM:.EXE\

                                                                                              .PSECT  $OWN$,NOEXE,2
```

```
        00# 00000 PRIBOO_DESCRIP:
                            .BYTE    0[3]                                               ;
        02  00003           .BYTE    2                                                  ;
        00# 00004           .BYTE    0[4]                                               ;
        00# 00008 LOADADR_DESCRIP:
                            .BYTE    0[3]                                               ;
        02  0000B           .BYTE    2                                                  ;
        00# 0000C           .BYTE    0[4]                                               ;
        00# 00010 PROMPT_DESCRIP:
                            .BYTE    0[3]                                               ;
        01  00013           .BYTE    1                                                  ;
            00014           .BLKB    4
        00# 00018 VBN_DESCRIP:
                            .BYTE    0[3]                                               ;
        02  0001B           .BYTE    2                                                  ;
        00# 0001C           .BYTE    0[4]                                               ;
            00020 PRIBOO_FAB:
                            .BLKB    80
            00070 BOOTBL_FAB:
                            .BLKB    80
            000C0 PRIBOO_FILNAM:
                            .BLKB    255
            001BF           .BLKB    1
            001C0 PRIBOO_EXP_NAME:
                            .BLKB    255
            002BF           .BLKB    1
        02  002C0 PRIBOO_NAM_BLK:
                            .BYTE    2                                                  ;
        60  002C1           .BYTE    96                                                 ;
        FF  002C2           .BYTE    -1                                                 ;
        00  002C3           .BYTE    0                                                  ;
  00000000' 002C4           .ADDRESS PRIBOO_FILNAM                                      ;
        00  002C8           .BYTE    0                                                  ;
        00  002C9           .BYTE    0                                                  ;
        FF  002CA           .BYTE    -1                                                 ;
        00  002CB           .BYTE    0                                                  ;
  00000000' 002CC           .ADDRESS PRIBOO_EXP_NAME                                    ;
  00000000  002D0           .LONG    0                                                  ;
      0000# 002D4           .WORD    0[8]                                               ;
      0000# 002E4           .WORD    0[3]                                               ;
      0000# 002EA           .WORD    0[3]                                               ;
  00000000  002F0           .LONG    0                                                  ;
  00000000  002F4           .LONG    0                                                  ;
        00  002F8           .BYTE    0                                                  ;
        00  002F9           .BYTE    0                                                  ;
        00  002FA           .BYTE    0                                                  ;
        00  002FB           .BYTE    0                                                  ;
        00  002FC           .BYTE    0                                                  ;
        00  002FD           .BYTE    0                                                  ;
        00# 002FE           .BYTE    0[2]                                               ;
  00000000  00300           .LONG    0                                                  ;
  00000000  00304           .LONG    0                                                  ;
  00000000  00308           .LONG    0                                                  ;
  00000000  0030C           .LONG    0                                                  ;
  00000000  00310           .LONG    0                                                  ;
  00000000  00314           .LONG    0                                                  ;
```

```
                    00000000# 00318                 .LONG    0[2]                                         ;
                          1D  00320 PRIBOO_XABFHC:
                                                    .BYTE    29                                           ;
                          2C  00321                 .BYTE    44                                           ;
                        0000  00322                 .WORD    0                                            ;
                    00000000  00324                 .LONG    0                                            ;
                    00000000# 00328                 .LONG    0[9]                                         ;
                              0034C PRIVILEGE_MASK:
                                                    .BLKB    8
                0204    0008  00354 GETJPI_ITEMLIST:
                                                    .WORD    8, 516                                       ;
                    00000000' 00358                 .ADDRESS PRIVILEGE_MASK                               ;
          00000000  00000000  0035C                 .LONG    0, 0                                         ;
                              00364 IO_STAT_BLOCK:
                                                    .BLKB    8
                              0036C TWO_BLOCK_BUF:
                                                    .BLKB    1024
                              0076C BOOTDEV_DESCRIP:
                                                    .BLKB    8
                              00774 BOOTDEV_CHAN:
                                                    .BLKB    2
                              00776                 .BLKB    2
                              00778 LOAD_ADR:
                                                    .BLKB    4
                    00000000# 0077C DEVCHAR_BUFF:
                                                    .LONG    0[3]                                         ;
                    0000000C  00788 DEVCHAR_DESCRIP:
                                                    .LONG    12                                           ;
                    00000000' 0078C                 .ADDRESS DEVCHAR_BUFF                                 ;
                         00#  00790 FILNAM_DESCRIP:
                                                    .BYTE    0[8]                                         ;
                    00000000  00798 FILSPEC_DESCRIP:
                                                    .LONG    0                                            ;
                    00000000' 0079C                 .ADDRESS PRIBOO_EXP_NAME                              ;
                              007A0 VBN:            .BLKB    4
                              007A4 STAT_BLOCK:
                                                    .BLKB    8
                          00  007AC YES_NO_BUF:
                                                    .BYTE    0                                            ;
                              007AD                 .BLKB    3
                    00000001  007B0 YES_NO_DESCRIP:
                                                    .LONG    1                                            ;
                    00000000' 007B4                 .ADDRESS YES_NO_BUF                                   ;
                    0000001A  007B8 LOGIO_DESCRIP:
                                                    .LONG    26                                           ;
                    00000000' 007BC                 .ADDRESS LOGIO_MSG                                    ;
                    00000011  007C0 VBN_BNDS_DESCRIP:
                                                    .LONG    17                                           ;
                    00000000' 007C4                 .ADDRESS VBN_BNDS_MSG                                 ;
                    0000004D  007C8 REMOUNT_DESCRIP:
                                                    .LONG    77                                           ;
                    00000000' 007CC                 .ADDRESS REMOUNT_MSG                                  ;
                    0000001C  007D0 NOTCONTIG_DESCRIP:
                                                    .LONG    28                                           ;
                    00000000' 007D4                 .ADDRESS NOTCONTIG_MSG                                ;
                    00000001  007D8 BRACKET_DESCRIP:
                                                    .LONG    1                                            ;
```

```
                                 00000000' 007DC          .ADDRESS ASCII_BRACKET                                    ;

                                                  BLOCK_BUFFER=         TWO_BLOCK_BUF
                                                  LOGIO_MSG=            P.AAA
                                                  VBN_BNDS_MSG=         P.AAB
                                                  NOTCONTIG_MSG=        P.AAC
                                                  REMOUNT_MSG=          P.AAD
                                                  ASCII_BRACKET=        P.AAE
                                                  PROMPT_BUFFER=        P.AAF
                                                  PROMPT2_BUFFER=       P.AAG
                                                  PROMPT3_BUFFER=       P.AAH
                                                  PRIBOO_DEF_NAME=      P.AAI
                                                  $RMS_PTR=             PRIBOO_FAB
                                                  $RMS_PTR=             BOOTBL_FAB
                                                        .EXTRN  OTS$CVT_TZ_L, LIB$INDEX
                                                        .EXTRN  LIB$PUT_OUTPUT, LIB$SFREE1_DD
                                                        .EXTRN  RTF$TARGET_DEV, RTF$OPENFILE
                                                        .EXTRN  LIB$GET_INPUT, SYS$GETJPI
                                                        .EXTRN  SYS$PARSE, SYS$GETDEV
                                                        .EXTRN  SYS$OPEN, SYS$CLOSE
                                                        .EXTRN  SYS$QIOW, SYS$ASSIGN

                                                        .PSECT  $CODE$,NOWRT,2

                                      0FFC 00000 WRITE_BOOT:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11       ; 0240
              5B 00000000G  00   9E 00002               MOVAB   SYS$CLOSE, R11                             :
              5A 00000000G  00   9E 00009               MOVAB   LIB$GET_INPUT, R10                         :
              59 00000000G  00   9E 00010               MOVAB   LIB$SFREE1_DD, R9                          :
              58 00000000G  00   9E 00017               MOVAB   LIB$PUT_OUTPUT, R8                         :
              57     0000'  CF   9E 0001E               MOVAB   PROMPT_DESCRIP, R7                         :
                            7E   7C 00023               CLRQ    -(SP)                                      ; 0295
                     0354   C7   9F 00025               PUSHAB  IO_STAT_BLOCK                              :
                     0344   C7   9F 00029               PUSHAB  GETJPI_ITEMLIST                            :
                            7E   7C 0002D               CLRQ    -(SP)                                      :
                            03   DD 0002F               PUSHL   #3                                         :
    00000000G  00           07   FB 00031               CALLS   #7, SYS$GETJPI                             :
               56           50   D0 00038               MOVL    R0, STATUS                                 :
               38           56   E9 0003B               BLBC    STATUS, 3$                                 :
                     033C   C7   95 0003E               TSTB    PRIVILEGE_MASK                             ; 0297
                            0B   19 00042               BLSS    1$                                         :
                     07A8   C7   9F 00044               PUSHAB  LOGIO_DESCRIP                              ; 0299
               68           01   FB 00048               CALLS   #1, LIB$PUT_OUTPUT                         :
               50           24   D0 0004B               MOVL    #36, R0                                    ; 0300
                                 04 0004E               RET                                                :
               67           35   B0 0004F 1$:           MOVW    #53, PROMPT_DESCRIP                        ; 0308
          04   A7    0000'  CF   9E 00052               MOVAB   PROMPT_BUFFER, PROMPT_DESCRIP+4            ; 0309
                       F0   A7   B5 00058               TSTW    PRIBOO_DESCRIP                             ; 0311
                            06   13 0005B               BEQL    2$                                         :
                       F0   A7   9F 0005D               PUSHAB  PRIBOO_DESCRIP                             ; 0312
               69           01   FB 00060               CALLS   #1, LIB$SFREE1_DD                          :
                       F0   A7   B5 00063 2$:           TSTW    PRIBOO_DESCRIP                             ; 0314
                            11   12 00066               BNEQ    4$                                         :
                            57   DD 00068               PUSHL   R7                                         ; 0317
                       F0   A7   9F 0006A               PUSHAB  PRIBOO_DESCRIP                             :
               6A           02   FB 0006D               CALLS   #2, LIB$GET_INPUT                          :
               56           50   D0 00070               MOVL    R0, STATUS                                 :
```

```
                                                          B  5
WRITEBOOT                                                 15-Sep-1984 23:36:40    VAX-11 Bliss-32 V4.0-742             Page  15
V04-000                                                   14-Sep-1984 11:58:06    [BOOTS.SRC]WRITEBOOT.B32;1                (3)

                                ED          56  E8 00073       BLBS    STATUS, 2$                                  ; 0318
                                          02AD  31 00076 3$:   BRW     24$                                         ; 0319
                                51     F0   A7  3C 00079 4$:   MOVZWL  PRIBOO_DESCRIP, R1                          ; 0327
                                50          01  CE 0007D       MNEGL   #1, COUNT                                   ; 0331
                                            15  11 00080       BRB     6$                                          ;
                           61   8F   F4 B740  91 00082 5$:     CMPB    @PRIBOO_DESCRIP+4[COUNT], #97               ; 0332
                                            0D  1F 00088       BLSSU   6$                                          ;
                           7A   8F   F4 B740  91 0008A         CMPB    @PRIBOO_DESCRIP+4[COUNT], #122              ;
                                            05  1A 00090       BGTRU   6$                                          ;
                           F4 B740          20  82 00092       SUBB2   #32, @PRIBOO_DESCRIP+4[COUNT]               ; 0333
                    E7          50          51  F2 00097 6$:   AOBLSS  R1, COUNT, 5$                               ; 0327
    0050   8F       00          6E          00  2C 0009B       MOVC5   #0, (SP), #0, #80, $RMS_PTR                 ; 0354
                                       10   A7     000A2                                                           ;
                           10   A7   5003   8F  B0 000A4       MOVW    #20483, $RMS_PTR                            ;
                           14   A7 01000000 8F  D0 000AA       MOVL    #16777216, $RMS_PTR+4                       ;
                           26   A7          02  90 000B2       MOVB    #2, $RMS_PTR+22                             ;
                           2F   A7          02  90 000B6       MOVB    #2, $RMS_PTR+31                             ;
                           34   A7   0310   C7  9E 000BA       MOVAB   PRIBOO_XABFHC, $RMS_PTR+36                  ;
                           38   A7   02B0   C7  9E 000C0       MOVAB   PRIBOO_NAM_BLK, $RMS_PTR+40                 ;
                           3C   A7     F4   A7  D0 000C6       MOVL    PRIBOO_DESCRIP+4, $RMS_PTR+44               ;
                           40   A7   0000'  CF  9E 000CB       MOVAB   PRIBOO_DEF_NAME, $RMS_PTR+48                ;
                           44   A7     F0   A7  90 000D1       MOVB    PRIBOO_DESCRIP, $RMS_PTR+52                 ;
                           45   A7          0F  90 000D6       MOVB    #15, $RMS_PTR+53                            ;
                                       10   A7  9F 000DA       PUSHAB  PRIBOO_FAB                                  ; 0355
                    00000000G   00          01  FB 000DD       CALLS   #1, SYS$PARSE                               ;
                                56          50  D0 000E4       MOVL    R0, STATUS                                  ;
                                7D          56  E9 000E7       BLBC    STATUS, 7$                                  ;
                         075C   C7   02C4   C7  9B 000EA       MOVZBW  PRIBOO_NAM_BLK+20, BOOTDEV_DESCRIP          ; 0358
                         0760   C7   02C5   C7  9E 000F1       MOVAB   PRIBOO_NAM_BLK+21, BOOTDEV_DESCRIP+4        ; 0359
                                            7E  7C 000F8       CLRQ    -(SP)                                       ; 0363
                                     0778   C7  9F 000FA       PUSHAB  DEVCHAR_DESCRIP                             ;
                                            7E  D4 000FE       CLRL    -(SP)                                       ;
                                     075C   C7  9F 00100       PUSHAB  BOOTDEV_DESCRIP                             ;
                    00000000G   00          05  FB 00104       CALLS   #5, SYS$GETDEV                              ;
                                56          50  D0 0010B       MOVL    R0, STATUS                                  ;
                                56          56  E9 0010E       BLBC    STATUS, 7$                                  ;
                                54   076F   C7  E9 00111       BLBC    DEVCHAR_BUFF+3, 8$                          ; 0380
                         0788   C7   02BB   C7  9B 00116       MOVZBW  PRIBOO_NAM_BLK+11, FILSPEC_DESCRIP          ; 0382
                                     07C8   C7  9F 0011D       PUSHAB  BRACKET_DESCRIP                             ; 0383
                                     0788   C7  9F 00121       PUSHAB  FILSPEC_DESCRIP                             ;
                    00000000G   00          02  FB 00125       CALLS   #2, LIB$INDEX                               ;
                         0784   C7   078C D740  9E 0012C       MOVAB   @FILSPEC_DESCRIP+4[INDEX], FILNAM_DESCRIP+4 ; 0384
        0780   C7        0788   C7          50  A3 00134       SUBW3   INDEX, FILSPEC_DESCRIP, FILNAM_DESCRIP      ; 0385
                                     075C   C7  9F 0013C       PUSHAB  BOOTDEV_DESCRIP                             ; 0387
                    00000000G   00          01  FB 00140       CALLS   #1, RTF$TARGET_DEV                          ;
                                     0794   C7  9F 00147       PUSHAB  STAT_BLOCK                                  ; 0389
                                     035C   C7  9F 0014B       PUSHAB  TWO_BLOCK_BUF                               ;
                                     0780   C7  9F 0014F       PUSHAB  FILNAM_DESCRIP                              ;
                    00000000G   00          03  FB 00153       CALLS   #3, RTF$OPENFILE                            ;
                                56          50  D0 0015A       MOVL    R0, STATUS                                  ;
                                53          56  E8 0015D       BLBS    STATUS, 12$                                 ;
                                     07B8   C7  9F 00160       PUSHAB  REMOUNT_DESCRIP                             ; 0393
                                68          01  FB 00164       CALLS   #1, LIB$PUT_OUTPUT                          ;
                                          01BC  31 00167 7$:   BRW     24$                                         ; 0394
                                       10   A7  9F 0016A 8$:   PUSHAB  PRIBOO_FAB                                  ; 0398
                    00000000G   00          01  FB 0016D       CALLS   #1, SYS$OPEN                                ;
                                56          50  D0 00174       MOVL    R0, STATUS                                  ;
```

```
                               ED              56  E9 00177        BLBC    STATUS, 7$                                  ;
                       0794    C7      0338    C7  D0 0017A        MOVL    PRIBOO_XABFHC+40, STAT_BLOCK                ; 0401
                                               13  12 00181        BNEQ    9$                                          ; 0402
                                       07C0    C7  9F 00183        PUSHAB  NOTCONTIG_DESCRIP                           ; 0404
                               68              01  FB 00187        CALLS   #1, LIB$PUT_OUTPUT                          ;
                                         10    A7  9F 0018A        PUSHAB  PRIBOO_FAB                                  ; 0405
                               6B              01  FB 0018D        CALLS   #1, SYS$CLOSE                               ;
                               50      02AC    8F  3C 00190        MOVZWL  #684, R0                                    ; 0406
                                                   04 00195        RET                                                 ;
                                       0324    C7  B5 00196 9$:    TSTW    PRIBOO_XABFHC+20                            ; 0408
                                               09  13 0019A        BEQL    10$                                         ;
                       0798    C7      0320    C7  D0 0019C        MOVL    PRIBOO_XABFHC+16, STAT_BLOCK+4              ; 0409
                                               08  11 001A3        BRB     11$                                         ;
        0798    C7     0320    C7              01  C3 001A5 10$:   SUBL3   #1, PRIBOO_XABFHC+16, STAT_BLOCK+4          ; 0410
                                         10    A7  9F 001AD 11$:   PUSHAB  PRIBOO_FAB                                  ; 0411
                               6B              01  FB 001B0        CALLS   #1, SYS$CLOSE                               ;
                               67              2D  B0 001B3 12$:   MOVW    #45, PROMPT_DESCRIP                         ; 0419
                         04    A7     0000'    CF  9E 001B6        MOVAB   PROMPT3_BUFFER, PROMPT_DESCRIP+4            ; 0420
                                               56  D4 001BC 13$:   CLRL    STATUS                                      ; 0422
                               4C              56  E8 001BE 14$:   BLBS    STATUS, 19$                                 ; 0423
                                         08    A7  B5 001C1        TSTW    VBN_DESCRIP                                 ; 0426
                                               06  13 001C4        BEQL    15$                                         ;
                                         08    A7  9F 001C6        PUSHAB  VBN_DESCRIP                                 ; 0427
                               69              01  FB 001C9        CALLS   #1, LIB$SFREE1_DD                           ;
                                               57  DD 001CC 15$:   PUSHL   R7                                          ; 0429
                                         08    A7  9F 001CE        PUSHAB  VBN_DESCRIP                                 ;
                               6A              02  FB 001D1        CALLS   #2, LIB$GET_INPUT                           ;
                               56              50  D0 001D4        MOVL    R0, STATUS                                  ;
                               30              56  E9 001D7        BLBC    STATUS, 18$                                 ;
                                         08    A7  B5 001DA        TSTW    VBN_DESCRIP                                 ; 0432
                                               13  13 001DD        BEQL    16$                                         ;
                                       0790    C7  9F 001DF        PUSHAB  VBN                                         ; 0433
                                         08    A7  9F 001E3        PUSHAB  VBN_DESCRIP                                 ;
                  00000000G    00              02  FB 001E6        CALLS   #2, OTS$CVT_TZ_L                            ;
                               56              50  D0 001ED        MOVL    R0, STATUS                                  ;
                                               05  11 001F0        BRB     17$                                         ;
                       0790    C7              01  D0 001F2 16$:   MOVL    #1, VBN                                     ; 0434
                                       0790    C7  D5 001F7 17$:   TSTL    VBN                                         ; 0436
                                               C1  14 001FB        BGTR    14$                                         ;
                                       07B0    C7  9F 001FD        PUSHAB  VBN_BNDS_DESCRIP                            ; 0439
                               68              01  FB 00201        CALLS   #1, LIB$PUT_OUTPUT                          ;
                               56              50  D0 00204        MOVL    R0, STATUS                                  ;
                               B2              56  E8 00207        BLBS    STATUS, 13$                                 ;
                                             0119  31 0020A 18$:   BRW     24$                                         ; 0440
                 50    0794    C7      0790    C7  C1 0020D 19$:   ADDL3   VBN, STAT_BLOCK, R0                         ; 0445
                       0794    C7        FF    A0  9E 00215        MOVAB   -1(R0), STAT_BLOCK                          ;
0050    8F       00                      6E    00  2C 0021B        MOVC5   #0, (SP), #0, #80, $RMS_PTR                 ; 0457
                                         60    A7     00222                                                        ;
                         60    A7      5003    8F  B0 00224        MOVW    #20483, $RMS_PTR                            ;
                         64    A7  00020000    8F  D0 0022A        MOVL    #131072, $RMS_PTR+4                         ;
                         76    A7              20  90 00232        MOVB    #32, $RMS_PTR+22                            ;
                         7F    A7              02  90 00236        MOVB    #2, $RMS_PTR+31                             ;
                       008C    C7     0000'    CF  9E 0023A        MOVAB   P.AAJ, $RMS_PTR+44                          ;
                       0090    C7     0000'    CF  9E 00241        MOVAB   P.AAK, $RMS_PTR+48                          ;
                       0094    C7      0F09    8F  B0 00248        MOVW    #3849, $RMS_PTR+52                          ;
                                         60    A7  9F 0024F        PUSHAB  BOOTBL_FAB                                  ; 0458
                  00000000G    00              01  FB 00252        CALLS   #1, SYS$OPEN                                ;
```

```
                56            50 D0 00259           MOVL    R0, STATUS                              ;
                AB            56 E9 0025C           BLBC    STATUS, 18$                             ;
                              7E 7C 0025F           CLRQ    -(SP)                                   ; 0472
                7E            01 7D 00261           MOVQ    #1, -(SP)                               ;
                7E     0200   8F 3C 00264           MOVZWL  #512, -(SP)                             ;
                       035C   C7 9F 00269           PUSHAB  BLOCK_BUFFER                            ;
                              7E 7C 0026D           CLRQ    -(SP)                                   ;
                7E            31 7D 0026F           MOVQ    #49, -(SP)                              ;
                         6C   A7 DD 00272           PUSHL   BOOTBL_FAB+12                           ;
                              7E D4 00275           CLRL    -(SP)                                   ;
   00000000G    00            0C FB 00277           CALLS   #12, SYS$QIOW                           ;
                56            50 D0 0027E           MOVL    R0, STATUS                              ;
                86            56 E9 00281           BLBC    STATUS, 18$                             ;
                67       41   8F 9B 00284           MOVZBW  #65, PROMPT_DESCRIP                     ; 0480
          04    A7    0000'   CF 9E 00288           MOVAB   PROMPT2_BUFFER, PROMPT_DESCRIP+4        ; 0481
                              56 D4 0028E           CLRL    STATUS                                  ; 0483
                3A            56 E8 00290  20$:     BLBS    STATUS, 23$                             ; 0485
                         F8   A7 B5 00293           TSTW    LOADADR_DESCRIP                         ; 0488
                              06 13 00296           BEQL    21$                                     ;
                         F8   A7 9F 00298           PUSHAB  LOADADR_DESCRIP                         ; 0489
                69            01 FB 0029B           CALLS   #1, LIB$SFREE1_DD                       ;
                              57 DD 0029E  21$:     PUSHL   R7                                      ; 0491
                         F8   A7 9F 002A0           PUSHAB  LOADADR_DESCRIP                         ;
                6A            02 FB 002A3           CALLS   #2, LIB$GET_INPUT                       ;
                56            50 D0 002A6           MOVL    R0, STATUS                              ;
                7A            56 E9 002A9           BLBC    STATUS, 24$                             ; 0492
                         F8   A7 B5 002AC           TSTW    LOADADR_DESCRIP                         ; 0494
                              13 13 002AF           BEQL    22$                                     ;
                       0768   C7 9F 002B1           PUSHAB  LOAD_ADR                                ; 0495
                         F8   A7 9F 002B5           PUSHAB  LOADADR_DESCRIP                         ;
   00000000G    00            02 FB 002B8           CALLS   #2, OTS$CVT_TZ_L                        ;
                56            50 D0 002BF           MOVL    R0, STATUS                              ;
                              CC 11 002C2           BRB     20$                                     ;
        0768    C7     0200   8F 3C 002C4  22$:     MOVZWL  #512, LOAD_ADR                          ; 0496
                              C3 11 002CB           BRB     20$                                     ; 0485
        035C    C7     0798   C7 D0 002CD  23$:     MOVL    STAT_BLOCK+4, BLOCK_BUFFER              ; 0504
        0360    C7     0796   C7 B0 002D4           MOVW    STAT_BLOCK+2, BLOCK_BUFFER+4            ; 0505
        0362    C7     0794   C7 B0 002DB           MOVW    STAT_BLOCK, BLOCK_BUFFER+6              ; 0506
        0364    C7     0768   C7 D0 002E2           MOVL    LOAD_ADR, BLOCK_BUFFER+8                ; 0507
                              7E 7C 002E9           CLRQ    -(SP)                                   ; 0517
                       0764   C7 9F 002EB           PUSHAB  BOOTDEV_CHAN                            ;
                       075C   C7 9F 002EF           PUSHAB  BOOTDEV_DESCRIP                         ;
   00000000G    00            04 FB 002F3           CALLS   #4, SYS$ASSIGN                          ;
                56            50 D0 002FA           MOVL    R0, STATUS                              ;
                26            56 E9 002FD           BLBC    STATUS, 24$                             ;
                              7E 7C 00300           CLRQ    -(SP)                                   ; 0529
                              7E 7C 00302           CLRQ    -(SP)                                   ;
                7E     0200   8F 3C 00304           MOVZWL  #512, -(SP)                             ;
                       035C   C7 9F 00309           PUSHAB  BLOCK_BUFFER                            ;
                              7E 7C 0030D           CLRQ    -(SP)                                   ;
                7E            20 7D 0030F           MOVQ    #32, -(SP)                              ;
                7E     0764   C7 3C 00312           MOVZWL  BOOTDEV_CHAN, -(SP)                     ;
                              7E D4 00317           CLRL    -(SP)                                   ;
   00000000G    00            0C FB 00319           CALLS   #12, SYS$QIOW                           ;
                56            50 D0 00320           MOVL    R0, STATUS                              ;
                04            56 E8 00323           BLBS    STATUS, 25$                             ;
                50            56 D0 00326  24$:     MOVL    STATUS, R0                              ; 0530
```

```
                                                 04 00329          RET                                           ;
                                   60       A7   9F 0032A 25$:     PUSHAB   BOOTBL_FAB                           ; 0536
                         6B                 01   FB 0032D          CALLS    #1, SYS$CLOSE                        ;
                         50                 01   D0 00330          MOVL     #1, R0                               ; 0542
                                                 04 00333          RET                                           ; 0543

; Routine Size:  820 bytes,    Routine Base:  $CODE$ + 0000


;  546          0544  1 END
;  547          0545  0 ELUDOM



;                          PSECT SUMMARY
;
;      Name                      Bytes                      Attributes
;
;  $OWN$                          2016  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                          351  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                          820  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                       Library Statistics
;
;                                         -------- Symbols --------      Pages       Processing
;      File                               Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619       82         0       1000          00:01.9



;                          COMMAND QUALIFIERS

;      BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:WRITEBOOT/OBJ=OBJ$:WRITEBOOT MSRC$:WRITEBOOT/UPDATE=(ENH$:WRITEBOOT)

; Size:          820 code + 2367 data bytes
; Run Time:         00:20.7
; Elapsed Time:     00:25.9
; Lines/CPU Min:     1577
; Lexemes/CPU-Min: 29927
; Memory Used:  276 pages
; Compilation Complete
```

0042 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CDDENTRY
LIS

CDD

CDDSHR
MAP

CDDEXC2
LIS

CDDLIB
LIS

CDDEXC
LIS

WRITEBOOT
LIS

CDDLIB
B32